ANALYTIX AUDIT

13/06/2022

PREMIUM AUDIT

NAME OF PROJECT	INME Run
TESTNET ADDRESS	0xCABa634307B94f04390afC144A9b841875289751
WEBSITE / TELEGRAM	https://www.inme.finance/

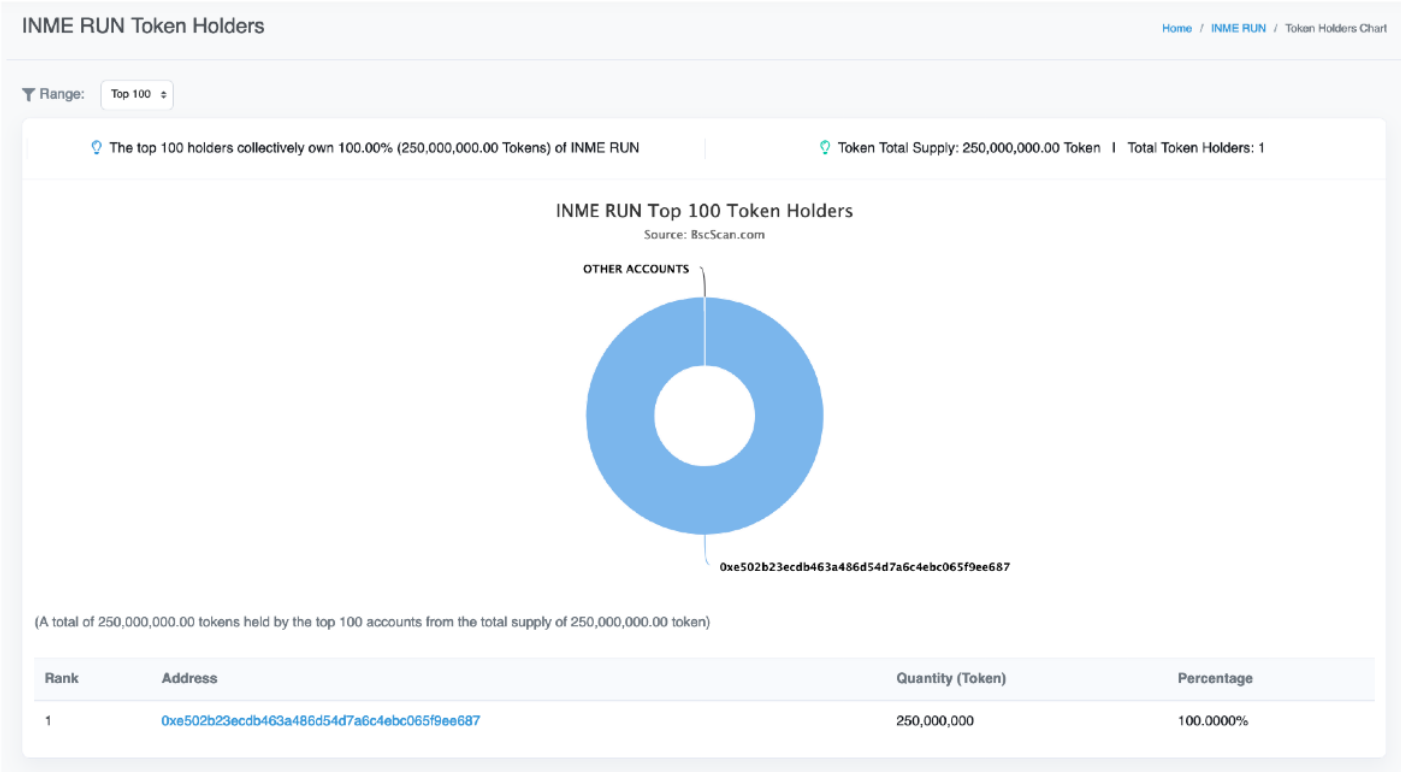
INME RUN

DISCLAIMER: PLEASE READ FULL AUDIT

IMPORTANT DISCLAIMER

I.	**NOT RESPONSIBLE**	Analytix Audit holds no responsibility for any actions from the project in this audit.
II.	**NOT GUARANTEE**	Analytix Audit in no way guarantees that a project will not remove liquidity, sell off team tokens, or exit scam.
III.	**INFORMATION**	Analytix Audit researches and provides public information about the project in an easy-to-understand format for the common person.
IV.	**AUDIT AGREEMENT**	This audit agreement does not guarantee ANY illicit actions by the project team and does not serve as an advocation for the project.
V.	**NO ADVICE**	Analytix Audit in no way takes responsibility for any losses, not does Analytix Audit encourage any speculative investments.
VI.	**DISCLAIMER**	The information provided in this audit is for information purposes only and should not be considered investment advice. Analytix Audit does not endorse, recommend, support, or suggest any projects that have been audited.

TOKENOMICS - OVERVIEW



INME RUN Token Holders

Home / INME RUN / Token Holders Chart

Range: Top 100

The top 100 holders collectively own 100.00% (250,000,000.00 Tokens) of INME RUN

Token Total Supply: 250,000,000.00 Token | Total Token Holders: 1

INME RUN Top 100 Token Holders

Source: BscScan.com

(A total of 250,000,000.00 tokens held by the top 100 accounts from the total supply of 250,000,000.00 token)

Rank	Address	Quantity (Token)	Percentage
1	0xe502b23ecdb463a486d54d7a6c4ebc065f9ee687	250,000,000	100.0000%

- More than 15% tokens unlocked

(This comment refers to when the audit was carried out maybe team locked tokens later)

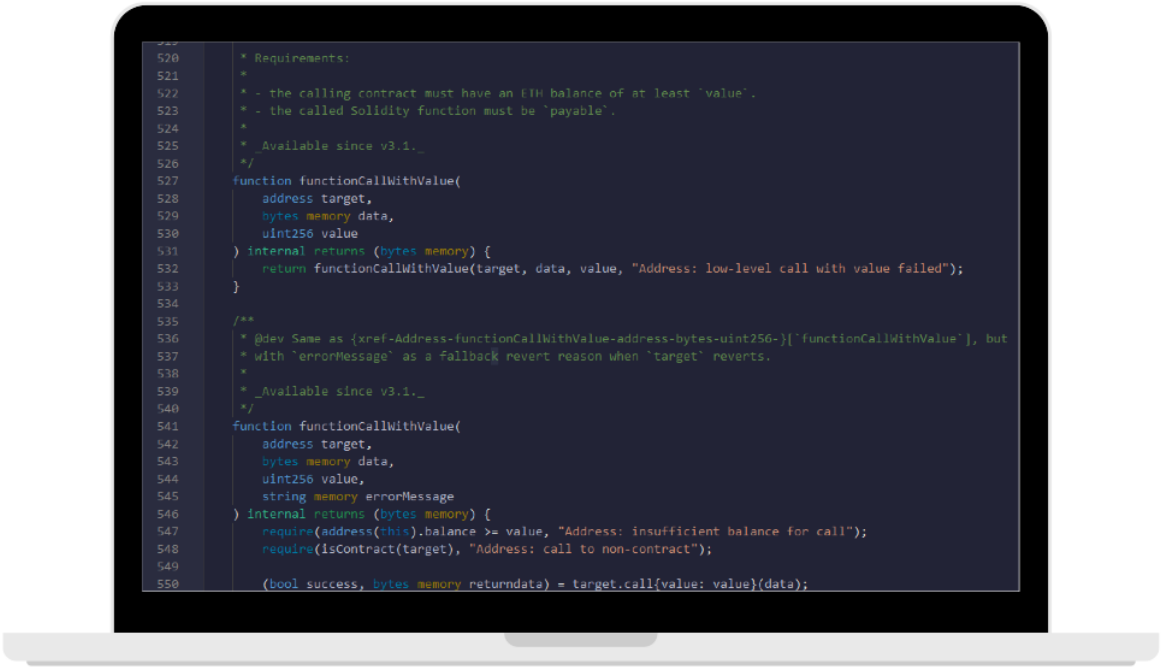
CONTRACT CODE - OVERVIEW

```
     * Requirements:
     *
     * - the calling contract must have an ETH balance of at least `value`.
     * - the called Solidity function must be `payable`.
     *
     * _Available since v3.1._
     */
    function functionCallWithValue(
        address target,
        bytes memory data,
        uint256 value
    ) internal returns (bytes memory) {
        return functionCallWithValue(target, data, value, "Address: low-level call with value failed");
    }

    /**
     * @dev Same as {xref-Address-functionCallWithValue-address-bytes-uint256-}[`functionCallWithValue`], but
     * with `errorMessage` as a fallback revert reason when `target` reverts.
     *
     * _Available since v3.1._
     */
    function functionCallWithValue(
        address target,
        bytes memory data,
        uint256 value,
        string memory errorMessage
    ) internal returns (bytes memory) {
        require(address(this).balance >= value, "Address: insufficient balance for call");
        require(isContract(target), "Address: call to non-contract");

        (bool success, bytes memory returndata) = target.call{value: value}(data);
```

QUICK REVIEW

- Owner can change buy/sell fees up to 20%
- Owner can set tx Limits
- Owner can't mint new tokens

Owner Privileges:



```
    * Requirements:
    *
    * - the calling contract must have an ETH balance of at least `value`.
    * - the called Solidity function must be `payable`.
    *
    * _Available since v3.1._
    */
    function functionCallWithValue(
        address target,
        bytes memory data,
        uint256 value
    ) internal returns (bytes memory) {
        return functionCallWithValue(target, data, value, "Address: low-level call with value failed");
    }

    /**
     * @dev Same as {xref-Address-functionCallWithValue-address-bytes-uint256-}[`functionCallWithValue`], but
     * with `errorMessage` as a fallback revert reason when `target` reverts.
     *
     * _Available since v3.1._
     */
    function functionCallWithValue(
        address target,
        bytes memory data,
        uint256 value,
        string memory errorMessage
    ) internal returns (bytes memory) {
        require(address(this).balance >= value, "Address: insufficient balance for call");
        require(isContract(target), "Address: call to non-contract");

        (bool success, bytes memory returndata) = target.call{value: value}(data);
```

function transferOwner(address newOwner) external onlyOwner {
function renounceOwnership() external onlyOwner {
function approveContractContingency() public onlyOwner returns (bool) {
function setNewRouter(address newRouter) public onlyOwner {
function setLpPair(address pair, bool enabled) external onlyOwner {
function setInitializers(address aInitializer, address cInitializer) external onlyOwner {
function setExcludedFromLimits(address account, bool enabled) external onlyOwner {
function setDividendExcluded(address holder, bool enabled) public onlyOwner {
function setExcludedFromFees(address account, bool enabled) public onlyOwner {
function setExcludedFromProtection(address account, bool enabled) external onlyOwner {
function removeSniper(address account) external onlyOwner {

(Owner can blacklist wallet)

function removeBlacklisted(address account) external onlyOwner {
function setProtectionSettings(bool _antiSnipe, bool _antiBlock) external onlyOwner {
function enableTrading() public onlyOwner {
function setWallets(address payable marketing, address payable liquidity, address payable competition) external onlyOwner {
function setTaxes(uint16 buyFee, uint16 sellFee, uint16 transferFee) external onlyOwner {
function setRatios(uint16 rewards, uint16 liquidity, uint16 marketing, uint16 competition) public onlyOwner {
function setMaxTxPercent(uint256 percent, uint256 divisor) external onlyOwner {
function setMaxWalletSize(uint256 percent, uint256 divisor) external onlyOwner {
function setSwapSettings(uint256 thresholdPercent, uint256 thresholdDivisor, uint256 amountPercent, uint256 amountDivisor) external onlyOwner {
function setPriceImpactSwapAmount(uint256 priceImpactSwapPercent) external onlyOwner {
function setContractSwapEnabled(bool swapEnabled, bool processReflectEnabled, bool priceImpactSwapEnabled) external onlyOwner {
function setRewardsProperties(uint256 _minPeriod, uint256 _minReflection, uint256 minReflectionMultiplier) external onlyOwner {
function setReflectorSettings(uint256 gas) external onlyOwner {
function excludePresaleAddresses(address router, address presale) external onlyOwner {
function multiSendTokens(address[] memory accounts, uint256[] memory amounts) external onlyOwner {
function sweepContingency() external onlyOwner {

Issue Checking:

N°	Issue description.	Checking Status.
1	Compiler Errors	Passed
2	Race conditions and Reentrancy. Cross-function race conditions.	Passed
3	Possible delays in data delivery.	Passed
4	Oracle calls.	Passed
5	Front running.	Passed
6	Timestamp dependence.	Passed
7	Integer Overflow and Underflow.	Passed
8	DoS with Revert.	Passed
9	DoS with block gas limit.	Passed
10	Methods execution permissions.	Passed
11	Economy model.	Passed
12	The impact of the exchange rate on the logic.	Passed
13	Private user data leaks.	Passed
14	Malicious Event log	Passed
15	Scoping and Declarations.	Passed
16	Uninitialized storage pointers.	Passed
17	Arithmetic accuracy.	Passed
18	Design Logic.	Passed
19	Cross-function race conditions.	Passed
20	Safe Zeppelin module	Passed
21	Fallback function securitv.	Passed

Audit Result:

PASSED

WEBSITE - OVERVIEW

Domain Registration: 2022-05-10

40 days old

SSL CERTIFICATE: A

(https://www.ssllabs.com)

SOCIAL MEDIA - OVERVIEW

INMERUN

INMEFINANCE

INME.FINANCE